

PMX1

OEM Software Build Guide

© COPYRIGHT VideoLogic Limited, 1998.

THE CONTENT OF THIS DOCUMENT IS PROPRIETARY TO VIDEOLOGIC LIMITED AND IS F
INTERNAL USE ONLY. IT SHOULD NOT BE COPIED OR ITS CONTENTS DISCLOSED WITHO
THE PRIOR CONSENT OF THE VIDEOLOGIC DIRECTOR RESPONSIBLE FOR PRODUCING TH
DOCUMENT.

Document ID : oembld1
Issue Number : c
Issue Date : 14th July 1998
Author : Dave Marsland



DOCUMENT HISTORY

Issue	Date	Changes / Comments
Draft A		Not Issued
Draft B	25 th June 1998	Added simulation & version details [RH].
Draft C	14 th July	Added Installation and SGL/HAL build instructions [RH].

**TABLE OF CONTENTS**

1.	Overview.....	1
2.	The Source Tree.....	1
3.	Building the overall tree.....	3
3.1.1	Development Tools.....	3
3.1.2	System set-up.....	3
3.1.3	Building the Full Win 9x Source	3
3.1.4	Building Specific Components	4
3.1.5	Runtime Environment.....	¡Error!Marcador no definido.
3.1.6	Win9x Installation.....	4
3.1.7	Versioning.....	5
3.2	Storing Internal Version Numbers	5
3.3	PMX Version Requirements	6



1. OVERVIEW

This document serves as a guide to the contents of the software source tree and how to build it.

2. THE SOURCE TREE

The root of the source tree is defined as \$(WORKROOT)\PMX\ and the following directories are relative to the PMX source tree root.

BINARY - built files end up here
 \bin_nt4
 \release
 \bin_w9x
 \release

CAPTURE - WDM
 COMMON
 BT829
 INC
 INSTALL
 VLDEBUG
 WDMDRV

DEBUG - Debug Vxd
 CLIENT
 DBGDRV
 WIN9X
 EXAMPLE
 APPS
 KERNMODE
 USERMODE
 UTILS

DIRECTX - DirectDraw, D3D
 COMMON
 D3DCOMMN
 DDCOMMN
 NT4
 WIN9X

DISPLAY - Display driver
 BIOS
 ASM
 INC
 TEST
 CPANEL
 WIN
 DRIVER
 INCLUDE



NT
WIN
MINIPORT

HWDEFS - hardware definitions

INF - file installation

PRODUCT - product versioning information

SERVICES - PMX device services

INCLUDE

PMXDXSRV - 3D services

COMMON

MODVER

NT4

NT5

WIN

PMXKERN - Kernel manager

COMMON

INCLUDE

NT4

NT5

WIN9X

PMXSERV - Display List Manager

COMMON

INCLUDE

NT4

NT5

WIN

SGLDIRECT - SGLDirect

BIN

COMMON

GLINFO

INCLUDE

PVR2OS

SGLPVR2

SGLTHIN

SWDEFS - shared software definitions

TOOLS

EXTERN

DDK

DX6

WDM

WIN

WIN95

WINNT

MASM

MSVC_16

MSVC_32



SDK
 DX6SDK
 WIN32
 SOFTICE

INTERN
 AUTOVER
 HEADERS

3. BUILDING THE OVERALL TREE

3.1.1 Development Tools

The following development tools are currently utilised. *Ensure you have valid licences before using these tools:-*

- Compiler MSVC 5.0
- Assembler MASM 6.12
- SDK's Microsoft Win 32, DX6
- DDK's Microsoft Windows 95/98 DDK, DX6
- make nmake
- Debugger Winice
- Utilities Norton File Date (FD)

3.1.2 System set-up

The following DOS Environment variables must be defined:

WORKROOT=[workpath]

[workpath] defines the root of your source tree e.g. c:\dev

SET LIB=[any setting] e.g. set LIB=.

SET INCLUDE=[any setting] e.g. set INCLUDE=.

All other internal environment settings are defined through incorporating *\$(WORKROOT)\pmx\swdefs\valenv.mk* into the makefiles.

3.1.3 Building the Full Win 9x Source

All make files operate from the DOS command prompt. To build the complete tree, from the root *\$(WORKROOT)\pmx* execute the command:

nmake -fpmxwin9x.mak <ChipName=1> RELEASE=1 target

Release Build

or



```
nmake -fpmxwin9x.mak <ChipName=1> DEBUG=1 target
```

Debug Build**where**

ChipName = PMX1C or PMX1LC

Binaries get built into the *\$(WORKROOT)\pmx\binary\bin_win9x\release* or *\$(WORKROOT)\pmx\binary\bin_win9x\debug* directories.

3.1.4 Building Specific Components

The Following components can be built individually from the top level makefile, using the following:

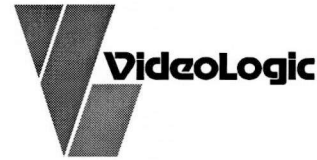
```
nmake PMX1C=1 [DEBUG=1] [RELEASE=1] -fpmxwin9x.mak component_name
```

where *component_name* is one of the following:

Component name	Driver component(s) built	Description
bios	Rombios.bin	VGA Bios
drv	Pmx.drv	Win9x display driver
mvxd	Pmxmini1.vxd, pmxmini2.vxd	Win9x display driver miniports
pmxkern	Pmxkern.vxd	Win9x kernel manager
pmxserv	Pmxserv.vxd	Win9x display list manager
pmxhal	Pmxhal.dll	Win9x, DX6 directX HAL
sgldirect	Sgl.dll, sgl2.dll, pvros.dll, sglmid6.dll, sglmid6b.dll	Win9x SGLDirect drivers
pmxcap	Pmxcap.sys	Win98 capture driver

3.1.5 Win9x Installation

To install the driver, use the display properties advanced setting option, and point installer to :
\$(WORKROOT)\pmx\binary\bin_w9x\release\pmx250i.inf



3.1.6 Versioning

With the exception of SGLDirect, all files are versioned according to the Microsoft numbering system.

The following table gives the range of version numbers appropriate for IHV- or OEM-supplied Windows 95 & 98 drivers for compatibility with various versions of DirectX.

Target system	Version number	
	Lowest number, including:	Up to, not including:
Windows 95-only drivers (no DirectX)	4.00.00.0095	4.02.00.0095
DirectX 1.0 compatible drivers	4.02.00.0095	4.03.00.1096
DirectX 2.0 compatible drivers	4.03.00.1096	4.03.00.2030
DirectX 3.0 compatible drivers	4.03.00.2030	4.04.00.0000
DirectX 5.0 and Windows 95 compatible drivers	4.10.01.0000	4.10.02.0000
DirectX 5.0 and Windows 98 compatible drivers	4.10.01.0000	4.10.02.0000
DirectX 6.x and Windows 9x compatible drivers	4.11.01.0000	

3.2 Storing Internal Version Numbers

In addition to the format that Microsoft requires for the version number, it is desirable to store an internal version number for product support and testing purposes. Every DirectX driver has one version number that is stored in duplicate: one binary version stored as two DWORDs, and one string version. **The binary version cannot be modified.** It is only the binary file version that is used at file installation. Time.

The string version, however, can be appended in the following way:

1. The vendor creates a version number, as described. This version number will be used "as is" in the binary version number.
2. The vendor uses this version number as the basis for the string version number. If desired, a vendor-specific version string can be appended to the existing version number to form the complete string version number. The vendor-specific string and the version number will be separated by a "-" (hyphen character).

For example, if "4.03.00.2100" is the version number for a DirectX-compliant display driver, and the vendor uses the "xx.xx.xx" number format internally, then the combined string version number in the driver will be "4.03.00.2100 -xx.xx.xx".

When the customer checks the version number of the driver (by selecting the file in Windows Explorer, choosing Properties, and then clicking the Version tab), Windows displays the string version. The vendor's product support should be able to identify the vendor-specific portion of the version number if it exists and take appropriate action.

3.3 PMX Version Requirements

All drivers and applications need to have a version attached to allow installation to work properly, aid technical support and problem tracking. Version data is located in the resource part of a Win32 executable. The version system used must comply with the Microsoft requirements outlined above.

There are 2 version numbers within the Microsoft Version resource; File Version and Product Version.

Both these version numbers are represented in a binary and a string form.

However, **the binary File Version** is the critical one used at installation time. The string value is not used.

The following numbering system will be used for DirectX compliant drivers:-

File Version Binary

Format:

4.11.01.xxxx

Where:

xxxx is an incrementing number – ideally the file build number (but needs to take into account branch numbers)

File Version String

Format:

4.11.01.xxxx-MA.MI.BG.xxxx

Where:

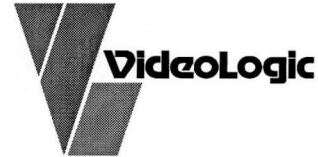
MA == Major change (e.g. to support completely new chip type)

MI == functional change (e.g. to support new features for an existing release)

BG == bug fix change (e.g. fix bug in existing release)

xxxx is the file build number

Example:



For PMX1-A1, the display driver, Ddraw, D3D & WDM drivers will have the file string versioned as:

4.11.01.xxxx-1.1.0.xxxx

File version strings are derived from individual project control files (PJ's). The PJ version number is used to represent the file version string, thus providing a direct means of tracing binaries back to source.

The utility *autover* is used to generate build numbers for each binary.

Product Versions (Binary and String)

MA.MI.BG.xxxx

Where:

MA == Major change (e.g. to support completely new chip type)

MI == functional change (e.g. to support new features for an existing release)

BG == bug fix change (e.g. fix bug in existing release)

xxxx is the Product build number and identifies the build within a release.

Example:

For PMX1-A1 the product version will be :

1.0.0.xxxx

The product version string xxxx reflects the build number and is obtained from PMX.PJ.